Joint Learning of Blind Video Denoising and Optical Flow Estimation

Songhyun Yu[1], Bumjun Park[1], Junwoo Park[2], and Jechang Jeong[1]
[1]Hanyang University, [2]Korea Advanced Institute of Science and Technology
fkdlzmftld@gmail.com, kkbbbj@gmail.com, junwoo.park@kaist.ac.kr, jjeong@hanyang.ac.kr

Abstract

Many deep-learning-based image/video denoising models have been developed, and recently, several approaches for training a denoising neural network without using clean images have been proposed. However, Noise2Noise method requires paired noisy data, and obtaining them is occasionally difficult, whereas other existing models trained using unpaired noisy data deliver limited performance. Obtaining an accurate optical flow from noisy videos is also a difficult task because conventional optical flow estimation methods are primarily focused on estimating the optical flow using clean videos. This study proposes a new framework to fine-tune video denoising and optical flow estimation networks using unpaired noisy videos. These two networks are jointly trained to realize synergy; an improvement in the denoising performance increases the accuracy of the flow estimation, and an improvement in the flow-estimation performance enhances the quality of the training data for the denoiser. Our experimental results reveal that proposed approach outperforms the existing training schemes in video denoising and also provides accurate optical flows even when the videos contain a considerable amount of noise.

1. Introduction

Advances in deep learning have considerably improved image- and video-denoising performances. Generally, convolutional neural network (CNN) is modeled to learn end-to-end mapping using noisy-clean training pairs, where the noisy images are generated by adding synthetic noise to clean images [1, 2, 3, 4]. In these methods, it is assumed that the noise in images is additive white Gaussian noise (AWGN).

However, owing to the in-camera image-processing pipeline and diversity of environment such as camera sensors, the distribution of real-world noise can be different from the Gaussian distribution, and a mismatch between the trained noise and tested noise can considerably degrade the denoising performance [5, 6, 7]. To address real noise, real-world datasets comprising images with real-world noise and the corresponding estimated ground-truth images were developed [8, 9, 10], and several training techniques have been studied to realize the training of a denoising network without using ground-truth images [6, 7, 11, 12, 13, 14].

Lehtinen et al. [14] developed an innovative approach to train an image denoising network, called Noise2Noise (N2N) training, from only noisy image pairs. Instead of learning a mapping from noisy images to clean targets, it was demonstrated that, in the case of particular noise types, a denoiser can be trained with only two independent noisy realizations of the same underlying clean image. In the case of real noise, however, the use of N2N training is limited because it is occasionally difficult to obtain two independent noisy images from the same underlying scene. To overcome this limitation, several methods [6, 7, 11, 12, 13] have been developed that are aimed at training a network using unpaired noisy images. Ehret et al. [6] proposed Frame2Frame (F2F) training that exploits the temporal redundancy in a video for training a denoising network. It comprises the use of optical flow estimation to obtain the noisy image pair from the same scene. However, the F2F training method has limitations and room for improvement that are described in detail in Section 3.1.

Optical flow estimation is a representative technique in computer vision, and it is widely used in many applications such as video-frame interpolation [15], action recognition [16], and surveillance systems [17]. Recently, deep-learning-based optical flow estimators have been actively developed [18, 19, 20], and they are trained on videos comprising ground-truth flows. However, these studies are primarily focused on clean sequences, and it is difficult to obtain an accurate optical flow from noisy sequences.

Inspired by N2N and F2F training, we design a practical framework to jointly fine-tune a denoising network and optical-flow-estimation network using unpaired noisy videos. Because an optical-flow estimator is fine-tuned on denoised frames and a denoiser is fine-tuned on training data generated using an optical flow estimation, the advancement of one network results in a performance improvement of the other network. Furthermore, we present an efficient warping method to improve the performance of N2N training from unpaired noisy videos.

2. Related Work

2.1. Image formation

Using a clean signal s and an additive noise a, we can formulate a noisy image as $n = s + a$. If we assume that the noise distribution has a zero mean and zero median, the following conditions are satisfied:

$$\mathbb{E}\{n\} = \mathrm{M}\{n\} = s \tag{1}$$

where $n = \{n^1, n^2, \cdots, n^k\}$ is a random variable of k independent noisy realizations obtained from the clean signal s. $\mathbb{E}\{\cdot\}$ and $\mathrm{M}\{\cdot\}$ denote the expectation and pixel-wise median, respectively. A representative noise that satisfies this property is the AWGN.

2.2. Supervised denoising

Given noisy-clean training pairs (n_i, s_i), the training procedure for supervised learning involves determining an optimal set of parameters as follows:

$$\underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{n,s}\{L(f_\theta(n), s)\} \tag{2}$$

where $f_\theta(\cdot)$ is the network function comprising the parameter set θ, and $L(\cdot)$ is a loss function. In general, L_1 loss and L_2 loss are used as the base loss for the image denoising [2, 4, 21].

2.3. Noise2Noise training

Given noisy-clean training pairs (n_i, s_i), the training of a neural network $f_\theta(\cdot)$ involves finding an optimal parameter set θ^* from (2). We can estimate the optimal network function as $f_{\theta^*}(n_i) = \mathbb{E}\{s_i|n_i\}$ for L_2 loss and $f_{\theta^*}(n_i) = \mathrm{M}\{s_i|n_i\}$ for L_1 loss [14]. Based on this property, Lehtinen et al. [14] demonstrated that, in the case of certain types of noise, the neural network can be trained using only noisy image pairs. For example, using L_2 loss, any data $\hat{s}_i$ satisfying

$$\mathbb{E}\{\hat{s}_i|n_i\} = \mathbb{E}\{s_i|n_i\} \tag{3}$$

can be used as target data instead of clean data. Similarly, data $\hat{s}_i$ satisfying

$$\mathrm{M}\{\hat{s}_i|n_i\} = \mathrm{M}\{s_i|n_i\} \tag{4}$$

can be used as target data to train the network based on L_1 loss. While obtaining ground-truth data is occasionally difficult and costly, this study eliminates the requirement of a clean image for training a denoising network. However, because the N2N method requires an independently generated noisy pair derived from the same underlying scene, its application is impractical in real-world image denoising.

2.4. Noise2Void training

To overcome the shortcomings of N2N training, Krull et al. [11] proposed a new scheme called Noise2Void (N2V) training for training a denoising network with unpaired noisy images. To construct a training data pair from a single noisy image, a blind spot network was introduced. If the receptive field centered on pixel p is $R(p)$, the network function can be described as follows:

$$f_\theta\left(n_{i,R(p)}\right) = s'_{i,p} \tag{5}$$

where $n_{i,R(p)}$ is a patch around pixel p of the noisy image n_i, and $s'_{i,p}$ is the output pixel at location p. The training data is interpreted as a receptive field-pixel pair in this approach. To prevent the network from learning identity mapping, several pixels are blinded in the noisy input patch. However, as this method employs only a few pixels in a patch to compute the loss, the training tends to be unstable, and it is difficult to set the optimal number of blind spots in a patch.

2.5. Frame2Frame training

Ehret et al. [6] proposed a training scheme for video denoising using unpaired noisy videos. By utilizing a high temporal redundancy of consecutive frames, this method warps adjacent frames to the target time using the optical flow. Therefore, it can construct noisy image pairs for N2N training from unpaired noisy videos. F2F training employs the TV-L1 method [22] to obtain optical flows in a noisy sequence owing to its high speed and robustness to noise [6]. If we denote the noisy frame at time t as n_t and the optical flow from time t to $t+1$ as $v_{t\to t+1}$, the data pair for N2N training is (n_t, n^w_{t+1}), where the warped frame at time t is represented as $n^w_{t+1}(p) = n_{t+1}(p + v_{t\to t+1}(p))$ with pixel position p. Therefore, using the warping method, we can create a dataset for N2N training from a single noisy sequence. A more detailed analysis of this method is presented in Section 3.1.

2.6. Optical flow estimation

Recently, many deep-learning-based methods for optical flow estimation have been studied [18, 20, 23, 24]. However, they are generally focused on noise-free sequences such as the KITTI [25] and Middlebury [26] datasets. In contrast, the final pass of the MPI Sintel dataset [27] includes camera noise, and FlowNet [18] is trained on augmented data with added Gaussian noise. However, these studies address noise of a relatively small intensity, and there exist few experiments on videos comprising severe noise. Using these pre-trained models on a severely contaminated video can result in critical performance degradation. While several non-deep-learning-based

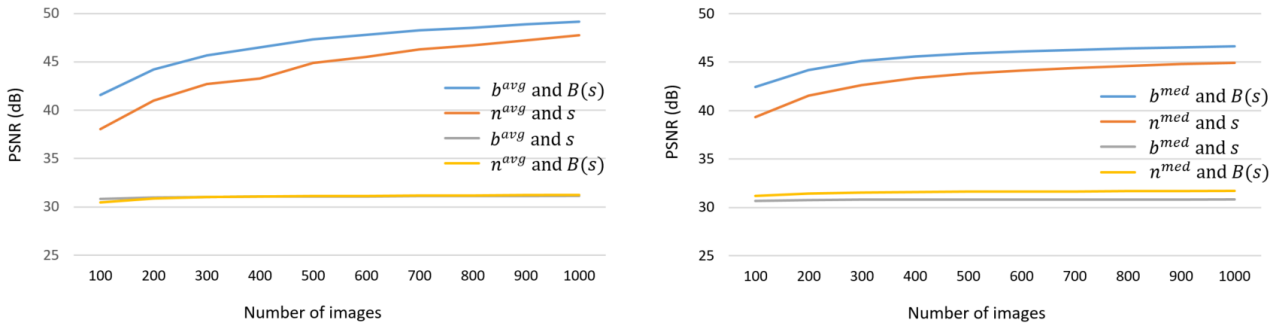


Figure 1: Mean and median values of filtered noisy images approach the filtered clean image and not the clean image, as the number of images increases.

studies have been conducted on optical flow in noisy sequences [28, 29], they cannot cope with some noise types, especially at high noise levels.

3. Proposed Methods

F2F training [6] forms the basis of the proposed framework; therefore, we first analyze its limitations and describe the details of the proposed framework in this section.

3.1. Limitations of Frame2Frame training

Although F2F training can be successfully used to develop a framework for training a denoising network from unpaired noisy videos, there exist limitations and room for improvement.

Interpolation method in warping. In F2F training, a noisy frame is warped to the target time using backward warping with bilinear interpolation. To successfully conduct N2N training based on L_2 loss, the distribution of the second noisy data $\hat{s}_i$ must follow Eq. (3). However, warping with bilinear interpolation negatively affects the statistical properties of the noisy signal that can be inferred based on simple intuition. We assume that noise is additive with a zero mean. If we fix the index i as j in the training dataset, Eq. (3) can be written as $\mathbb{E}\{\hat{s}_j|n_j\} = \mathbb{E}\{s_j|n_j\} = s_j$, where s_j is a fixed clean target at index j. As bilinear interpolation is the weighted sum of the surrounding pixels, we can approximate it as a type of linear smoothing filter (e.g., a 2D Gaussian filter). Therefore, by linearity of conditional expectation,

$$\mathbb{E}\{B(\hat{s}_j)|n_j\} = B(\mathbb{E}\{\hat{s}_j|n_j\}) = B(s_j), \tag{6}$$

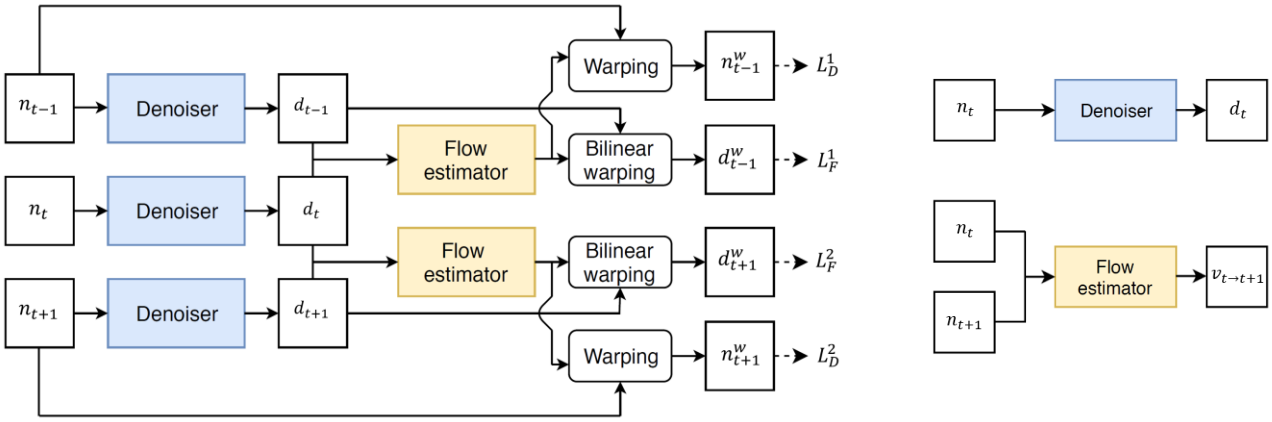
where $B(\cdot)$ is the linear smoothing function; the expected value of the smoothed noisy images is the smoothed clean image. Therefore, this property can be extended for an arbitrary index i, and this change in the expected value causes a quality degradation in the training data in the case of N2N training. Similarly, it can be inferred that bilinear interpolation can have an effect on the median value of the noisy signal, which violates Eq. (4) and causes the degradation of the N2N training based on L_1 loss.

To empirically verify this property, we conduct a toy experiment as detailed in Figure 1. For a clean image s, we create N noisy images with independent realizations, $n^1, n^2, \cdots, n^N$, using AWGN. The filtered images of each noisy image using a 2D Gaussian filter are represented as $b^1, b^2, \cdots, b^N$. Let us denote the average and pixel-wise median values of N noisy images as n^{avg} and n^{med}, respectively, and the corresponding values of the filtered images as b^{avg} and b^{med}, respectively. Figure 1 presents the peak signal-to-noise ratio (PSNR) between images according to N, and it demonstrates that as N increases, n^{avg} and n^{med} approach s, and b^{avg} and b^{med} approach $B(s)$, where $B(\cdot)$ represents filtering function of 2D Gaussian. To train a network using the N2N method, b^{avg} and b^{med} must approach s instead of $B(s)$ as N increases; however, as illustrated in Figure 1, their PSNR does not increase further.

Fixed flow estimator. TV-L1 [22], which is an optical-flow estimator that is robust against noise, is used in F2F training. However, its estimation accuracy is degraded depending on the levels and types of noise, especially at high noise levels. Because the performance of the flow estimator is directly related to the performance of the denoising network, selecting an estimator to fit the noise types and levels can improve the denoising performance. Our proposed method computes the optical flow from the denoised frames and is fine-tuned for target levels and types of noise. In this manner, the performance of a denoising network is improved, and an accurate optical flow can also be obtained from given noisy videos.

3.2. Joint learning of blind video denoising and optical flow estimation

Our objective is to obtain denoised video and optical flow from noisy videos at the same time without using any ground-truth in given noisy type. We improve upon the F2F training method, which is focused on video denoising using a non-trainable optical flow estimator: we not only enhance



(a) Training phase (b) Test phase

Figure 2: Overall flow chart of the proposed framework; Our framework consists of one denoising network and one flow-estimation network, and these are jointly trained. In the test phase, the two networks are used independently because the flow estimator takes noisy frames directly as the input.

the denoising performance, but also obtain an accurate optical flow by fine-tuning the trainable estimator from unpaired noisy videos. To realize stability and a high convergence speed, we use pre-trained models as the initial parameters: DnCNN [1] and PWC-Net [24] are used as the denoising and flow-estimation networks, respectively. DnCNN is pre-trained using AWGN with $\sigma \in [0, 50]$ on DIV2K [39] dataset, and PWC-Net is pre-trained using a MPI Sintel [40] dataset with ground-truth optical flow.

Figure 2 presents the overall flow chart of the proposed framework that comprises a denoiser and a flow estimator, and Figure 2(a) illustrates the training phase. First, given three consecutive noisy frames n_{t-1}, n_t, and n_{t+1}, the denoiser generates denoised frames of each, i.e., d_{t-1}, d_t, and d_{t+1}, respectively. Second, the flow estimator computes the bi-directional flows $v_{t-1\to t}$ and $v_{t+1\to t}$ from denoised frames. Using these optical flows, noisy targets are generated as follows:

$$n^w_{t-1}(p + R(v_{t-1\to t}(p))) = n_{t-1}(p), \tag{7}$$
$$n^w_{t+1}(p + R(v_{t+1\to t}(p))) = n_{t+1}(p), \tag{8}$$

where $R(\cdot)$ is a rounding operation. Therefore, the loss function for training the denoiser is given as follows:

$$\begin{gathered} L_D = L^1_D + L^2_D, \\ \text{where } L^1_D = \textstyle\sum_i A(n^w_{t-1}(i))|d_t(i) - n^w_{t-1}(i)| \\ \text{and } L^2_D = \textstyle\sum_i A(n^w_{t+1}(i))|d_t(i) - n^w_{t+1}(i)|, \end{gathered} \tag{9}$$

$$A(x) = \begin{cases} 1, \text{ if } x \text{ is not hole} \\ 0, \text{ otherwise,} \end{cases} \tag{10}$$

where $A(\cdot)$ is a masking function used to exclude holes from the loss computation. The noisy frames are warped using forward warping without using any interpolation process, which has two advantages. Firstly, the noisy frames maintain their data distribution during warping, and they are used as high-quality noisy targets. Secondly, the warped frames have holes where optical flows are not assigned, and these holes serve as a mask for filtering out occlusion and inaccurate motions. While F2F computes occluded regions and masks them in the loss calculation, it is not necessary to compute occluded regions in our method. To train the flow estimator, we warp the *denoised* frames as follows:

$$d^w_{t-1}(p) = d_{t-1}(p + v_{t\to t-1}(p)), \tag{11}$$
$$d^w_{t+1}(p) = d_{t+1}(p + v_{t\to t+1}(p)), \tag{12}$$

where bilinear interpolation is used. Therefore, the loss function for the flow estimator is given as follows:

$$\begin{gathered} L_F = L^1_F + L^2_F, \\ \text{where } L^1_F = \textstyle\sum_i |d_t(i) - d^w_{t-1}(i)| \\ \text{and } L^2_F = \textstyle\sum_i |d_t(i) - d^w_{t+1}(i)|. \end{gathered} \tag{13}$$

In contrast to the training denoiser, bilinear interpolation is used here because it is not necessary to maintain the statistical property of the data. In the test phase, our flow estimator takes noisy frames directly as inputs. Therefore, the two networks (i.e., denoiser and flow estimator) are trained together, but they can be used independently in the test phase as shown in Figure 2(b).

To train/test the flow estimator, we have to decide whether to use denoised frames or noisy frames at each step, and there are several possible cases as listed in Table 1. Let us simplify denoised frames as 'D' and noisy frames as 'N', then the last row of Table 1 is denoted as 'D-D-N'. We can

Training Input	Training Loss	Test	PSNR
Denoised	Noisy	Denoised	30.05
		Noisy	30.14
Noisy	Noisy	Denoised	29.76
		Noisy	30.28
Noisy	Denoised	Denoised	30.88
		Noisy	31.15
Denoised	Denoised	Denoised	30.84
		Noisy	**31.99**

Table 1: Performance comparison by frame selection at each step. Validation PSNR (dB) is measured on AWGN.

summarize and analyze the results of Table 1 as follows. 1) Under the same training conditions, use of denoised frames in the test phase significantly degrades the performance because we use an image denoising network that employs single input image, which does not maintain temporal consistency of consecutive denoised frames. If we use a video denoising network that preserves temporal consistency, 'D-D-D' scheme outperforms 'D-D-N', which supports the above analysis. Video version of the proposed framework is discussed in Section 4.5. 2) Using noisy frames for the loss function destabilizes the training and degrades performance. This may be because noise occurs independently in consecutive frames; therefore, smoothness and convexity of the loss function could be damaged. 3) 'N-D-N' is the second best choice, but it is still inferior to 'D-D-N'. It can be inferred that the mismatched domains of input and loss hinder training. Consequently, our flow estimator employs denoised frames in the training phase and noisy frames in the test phase (D-D-N).

If the noise distribution used in our experiment follows Eqs. (3) and (4), it is possible to train a denoiser based on L_1 loss and L_2 loss. As demonstrated in [32], L_1 loss delivers a superior training performance than L_2 loss in image restoration. Therefore, all our networks are trained using L_1 loss as in [6].

4. Experimental Results

In this section, we detail the training strategy, ablation study, and comparison results with existing training schemes. As we focus on the training method itself and not on the model structure, all the presented denoising schemes are experimented using the same network structure.

4.1. Training details and dataset

Our denoiser and flow estimator are jointly trained to realize synergy. There are several possible options for training two networks simultaneously. We train them one by one: the denoiser is trained first using a fixed pre-trained flow estimator; the flow estimator is then trained by fixing the trained denoiser, and the denoiser is again fine-tuned using trained weights of the flow estimator. Figure 3 shows the training progress of the two networks. In the second training phase of the denoiser, the performance improves drastically owing to the improved flow-estimation performance.

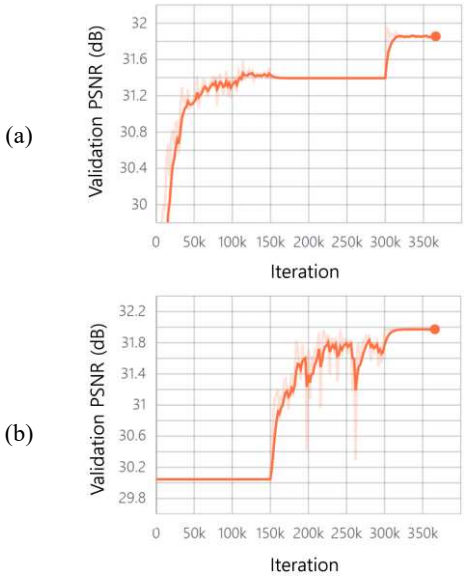


Figure 3: Training progress of (a) denoiser and (b) flow estimator; Two networks are trained alternately. Fine-tuned flow estimator improves the denoiser performance.

For the training, we use the Vimeo-90K dataset [30], which is a widely used dataset for video-processing research. The training images are not divided into patches; instead, entire images of 448 × 256 resolution are used as the training data with a batch size of 1. The Adam optimizer [33] is used for training with an initial learning rate of $1e^{-4}$, which is halved after every 30K iterations. The denoiser is trained for 150K iterations, after which the flow estimator is trained for 150K iterations, and the denoiser is then fine-tuned for another 60K iterations. Approximately 24 h are required to train a single framework (denoiser and flow estimator) using RTX 2080Ti.

4.2. Ablation study

In this section, we present the ablation results. The results are compared in terms of image denoising and optical flow estimation using AWGN with a σ value of 50 as listed in Table 2.

Table 2(a) compares the effects of the flow estimators. Bi-directional optical flows give gain of denoising and flow estimation performances by handling inaccuracies of motion estimation and occlusion. The use of SpyNet [23], which delivers inferior performance in optical-flow benchmarks than PWC-Net [24], reduces the accuracy of

Methods	Denoising	Flow estimation
Uni-direction	26.47	28.30
SpyNet	26.55	27.26
Proposed method	**26.56**	**28.35**

(a) Variations of flow estimator; Bi-directional warping improves performances as compared to uni-directional warping. Using SpyNet instead of PWC-Net decreases the accuracy of the flow estimator; however, it does not significantly affect the denoising performance.

Methods	Denoising	Flow estimation
Flow-Den	26.32	28.26
Flow&Den	26.27	28.28
Proposed method	**26.56**	**28.35**

(b) Effect of training order; Training a denoiser followed by the flow estimator (Proposed) is the superior training order for the two networks than training the flow estimator first followed by the denoiser (Flow-Den) and training the two networks simultaneously from the start (Flow-&Den).

Methods	Denoising	Flow estimation
Bilinear	25.35	27.83
Nearest	26.49	28.27
Proposed method	**26.56**	**28.35**

(c) Effect of interpolation methods in warping; Bilinear interpolation severely degrades the final results, and the nearest neighbor interpolation shows results that are comparable to that of the proposed scheme.

Table 2: Ablation results in AWGN with $\sigma = 50$.

the flow estimation; however, it does not significantly degrade the denoising performance. Thus, a denoiser in our training scheme is robust to the poor performance of the flow estimator, and the use of other flow estimators that deliver better performance than PWC-Net can potentially improve the flow-estimation accuracy in noisy videos.

Table 2(b) explains the effects of the training order. Reversing the order of training of the two networks (Flow-Den) degrades the performances of both the networks. If both the networks are trained simultaneously from the start (Flow&Den), they affect each other in unstable states. In particular, the initial image denoiser suffers from poor temporal consistency of denoised frames, which degrades the final performances of both the networks. Therefore, we first stabilize a denoising network with the target noise and then train the flow estimator.

Table 2(c) presents the effects of interpolation methods in warping. As mentioned in Section 3.1, warping with bilinear interpolation significantly reduces the performances even in flow estimation. In contrast, the nearest neighbor interpolation yields results that are comparable to those of the proposed method because it does not smooth the noisy images.

4.3. Experimental setting of conventional methods

For a fair comparison, all the networks in the conventional methods are trained using pre-trained DnCNN_B as initial weights with the DIV2K dataset [39], where _B indicates the blind version of the AWGN. All networks are trained based on L_1 loss.

V-BM4D [42]. Open source code is used with default settings. Noise estimation is used to assume blind denoising.

N2N. A training dataset is constructed by independently adding same type of noise twice to each clean image.

N2V. In the original models in [11], the authors have experimented with gray-scale images while using U-Net [34] as the basic network structure. In our experiment, N2V is trained on RGB images based on the DnCNN structure and using patches of size 200 × 200 with a batch size of 1. The initial learning rate is set as $5e^{-4}$, which is halved after every 60K iterations.

F2F. A modified TV-L1 algorithm is used in the previous study [6], and in public implementation, the optical flow of a target sequence is obtained in advance of training a denoiser. We modified the aforementioned implementation to compute the flow in real time for training a generalized model and used the TV-L1 method built in the openCV library. Because TV-L1 uses gray-scale images as the input, we converted RGB frames to gray-scale when computing the optical flow.

4.4. Comparison with conventional methods

Models trained using the proposed framework and conventional training schemes are compared in this section. Denoising and optical flow estimation experiments have been conducted using the Vid4 [35] and DAVIS [36] datasets, respectively. Vid4 consists of four sequences, and DAVIS consists of 30 sequences including a variety of motions. To evaluate the accuracy of optical flows in a test sequence without ground-truth flows, we compute the PSNR of the warped clean images and ground-truth images. Three noise models are used for the evaluation: AWGN, Poisson-Gaussian noise, and speckle noise. Tables 3 and 4 list the obtained results. "DnCNN_S" represents the conventional supervised training results with a clean target, and "DnCNN_B" indicates the blind version of DnCNN model trained on AWGN with $\sigma \in [0, 50]$. In Table 4, all the methods except TV-L1 employ pre-trained PWC-Net, and it is fine-tuned using different strategies. "f.t. – clean video" means fine-tuned and tested in noise-free videos, "f.t. – noisy video" means fine-tuned and tested in noisy videos, and "f.t. – DnCNN_B" means fine-tuned in denoised videos using pre-trained DnCNN_B.

Gaussian noise. The methods are compared for AWGN at standard deviations of 25 and 50. Table 3 indicates that the proposed method delivers the best denoising performance (i.e., more than a 1-dB increase over that of the F2F method for $\sigma = 50$). F2F performs well at low noise levels, but at high noise levels, PSNR decreases because the flow-estimation accuracy of TV-L1 is significantly degraded at a high noise level. Compared to supervised learning and the N2N method, the proposed method exhibits a competitive performance, although our network is trained using unpaired noisy videos. Table 4 indicates that the proposed method delivers the highest flow-estimation accuracies among the fine-tuning schemes and TV-L1 method.

Poisson-Gaussian noise. We experiment with Poisson-Gaussian noise modeling as in [41], which is given by:

$$n(p) = s(p) + \eta_o\big(s(p)\big) + \eta_g(p), \tag{14}$$

where η_o is a signal-dependent Poisson component, and η_g is a signal-independent Gaussian component. Standard deviation of the Gaussian noise is set to 10 to assume low-level noise conditions. Tables 3 and 4 indicate that the proposed method outperforms all conventional methods.

Speckle noise. Speckle noise is a multiplicative noise that can be expressed as $n = sa$, where a is a uniformly distributed noise with a specified mean and variance. When the data range is [0, 1], we set the variance as 0.5 to assume a severe noisy environment, where the mean is set as zero to satisfy Eq. (4). Table 3 indicates that the proposed method delivers the highest denoising performance. The performance of F2F is inferior to that of N2V; this demonstrates that F2F suffers from poor flow-estimation accuracy of the TV-L1 algorithm under severe noise conditions as indicated by Table 4. In Table 4, our flow estimator significantly outperforms the other fine-tuning schemes. Because DnCNN_B is pre-trained on AWGN, the fine-tuned estimator that employs the fixed DnCNN_B performs well with Gaussian noise; however, in the case of noise having different characteristics, the denoised frames are degraded, and the performance of the fine-tuned flow estimator is affected.

Figure 4 presents the denoising results in the case of speckle noise. Although we use a pre-trained denoiser with AWGN, clear images that are comparable to the results obtained with supervised learning and N2N [14] training are observed. Figure 5 illustrates the warped frames at speckle noise. The proposed fine-tuned flow estimator computes an accurate flow to increase the warped PSNR even in a severe noise condition.

4.5. Video denoising network

As mentioned in Section 3.2, the performance of our flow estimator decreases when using denoised frames as inputs.

Methods	AWGN		Poisson-Gaussian	Speckle
	$\sigma = 25$	$\sigma = 50$		
V-BM4D [42]	29.13	26.31	32.08	18.87
DnCNN_S [1]	30.13	26.64	33.18	25.36
N2N [14]	30.03	26.67	33.16	25.38
DnCNN_B [1]	29.47	24.65	32.61	22.09
N2V [11]	27.81	25.17	29.66	24.10
F2F [6]	29.59	25.46	32.36	23.08
Proposed method	**29.96**	**26.56**	**32.70**	**24.81**

Table 3: Denoising results for three types of noise in Vid4 [35] dataset (PSNR (dB)).

Methods	AWGN		Poisson-Gaussian	Speckle
	$\sigma = 25$	$\sigma = 50$		
f.t. – clean video	30.55	30.55	30.55	30.55
TV-L1	26.27	25.68	26.14	23.73
PWC-Net [24]	26.33	26.28	26.26	25.51
f.t. – noisy video	28.49	25.26	29.41	21.98
f.t. – DnCNN_B	29.93	28.26	30.25	25.71
Proposed method	**30.12**	**28.35**	**30.53**	**27.05**

Table 4: Optical-flow-estimation results for three types of noise in DAVIS [36] dataset; PSNRs are measured between warped clean image and ground truth.

We assumed that this is due to the poor temporal consistency of the image-denoising network. We conduct an experiment with a video-denoising network to confirm this hypothesis. Video denoising networks typically use multiple frames as input and maintain a good temporal consistency [37]. We modify a simple U-Net [34] such that it employs three frames as the input and outputs an intermediate frame. Generally, video-processing networks take successive frames as input. However, because we obtain the target data from adjacent frames, using them again as input would introduce the risk of a denoiser learning just warping. Therefore, we design a video-denoising network as follows:

$$\begin{aligned} f_\theta^v(n_{t-2}, n_t, n_{t+2}) &= d_t, \\ L_D^v &= \textstyle\sum_i A(n_{t+1}^w(i))|d_t(i) - n_{t+1}^w(i)|, \\ L_F^v &= \textstyle\sum_i |d_t(i) - d_{t+1}^w(i)|. \end{aligned} \tag{15}$$

Instead of using adjacent frames n_{t-1} and n_{t+1}, we use frames n_{t-2} and n_{t+2} as the additional input. In the case of AWGN with a σ of 50, our video-denoising network achieves a PSNRs of denoising and flow estimation that are approximately 0.2 dB higher than that achieved by the image-denoising network. Owing to good temporal consistency, the PSNR improvement in the flow estimation is caused by using denoised frames as inputs in the test phase. If high-performance networks are designed for our

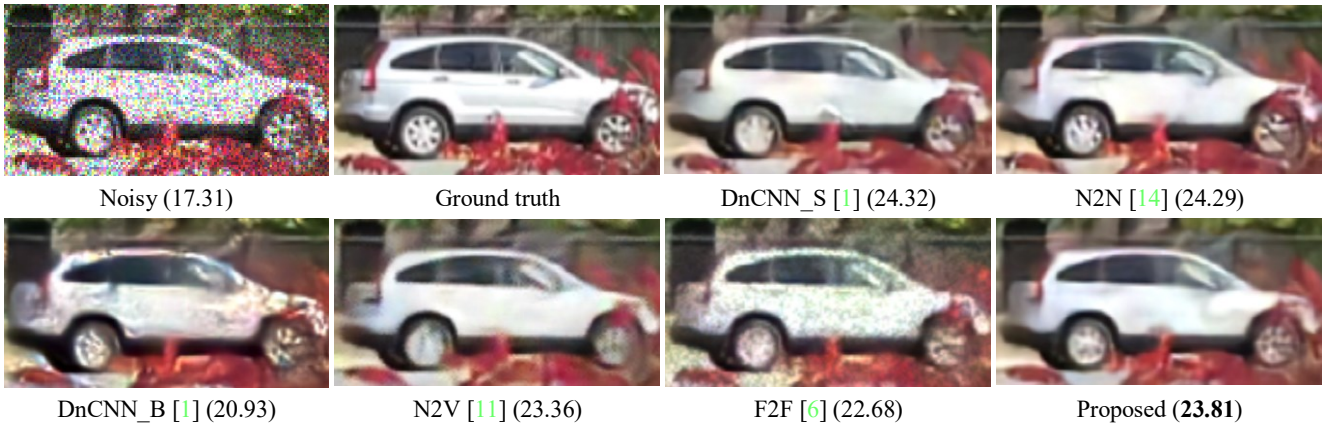


Figure 4: Denoised frames at Vid4 dataset with speckle noise (PSNR (dB)).

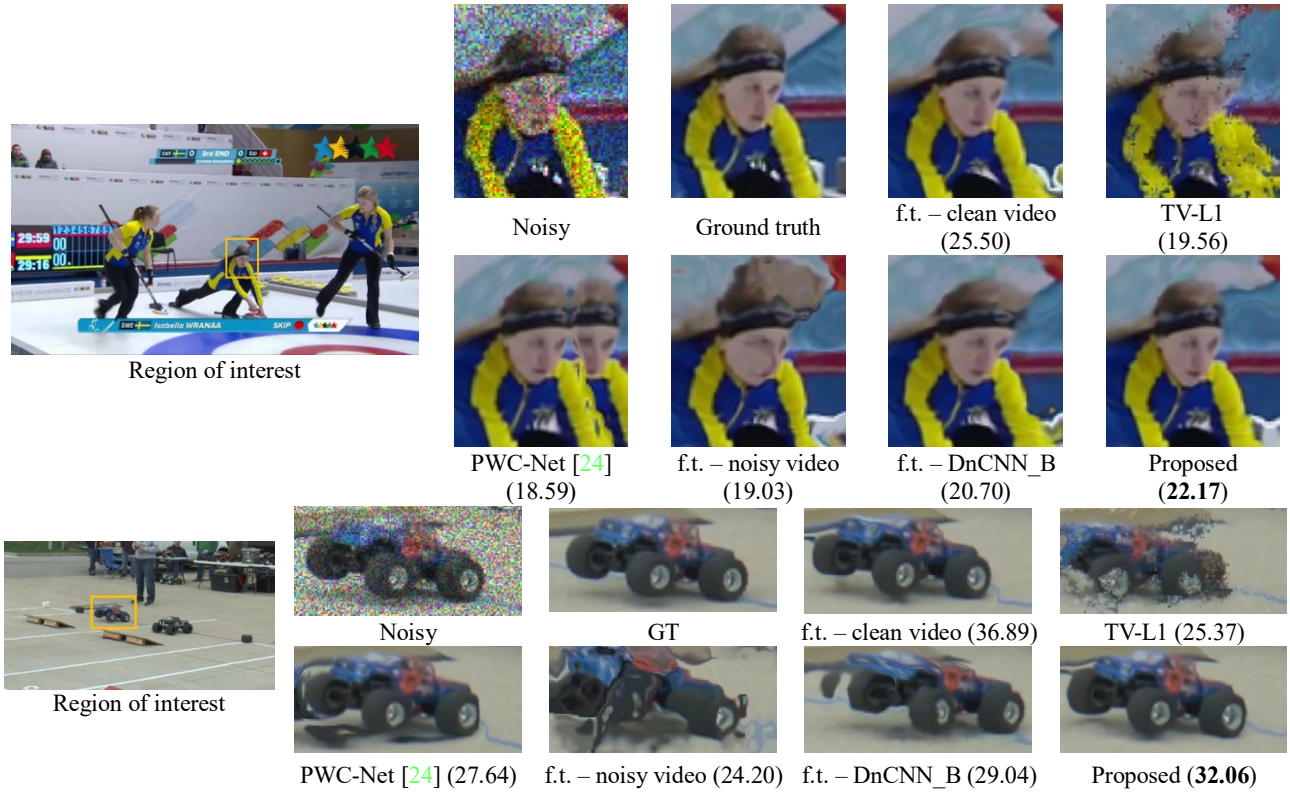


Figure 5: Warped frames at DAVIS dataset with speckle noise (PSNR (dB)).

video-denoising scheme, the realized performances could potentially be further improved.

5. Conclusion

In this paper, we presented a framework to fine-tune networks for video denoising and optical-flow estimation simultaneously from unpaired noisy videos. In the existing F2F training, the denoising network is trained using an fixed optical flow; we proposed a method to improve the performance of both the networks by jointly training them. In addition, using forward warping without an interpolation process, we successfully improved the denoising performance. In the test phase, our denoiser and flow estimator can be used independently because the flow estimator directly takes noisy frames as the input. Furthermore, we presented that a network designed for video-denoising significantly enhances the denoising and flow-estimation performances. The experimental results demonstrated that unpaired noisy videos are sufficient for obtaining considerably good results in video denoising and optical flow estimation.

References

[1] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing,* 26(7):3142–3155, 2017.

[2] Y. Tai, J. Yang, X. Liu, and C. Xu. MemNet: A persistent memory network for image restoration. In *ICCV*, pages 4539–4547, 2017.

[3] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo. Multi-level wavelet-CNN for image restoration. In *CVPR*, pages 773–782, 2018.

[4] S. Yu, B. Park, and J. Jeong. Deep iterative down-up CNN for image denoising. In *CVPRW*, 2019.

[5] K. Lin, T. H. Li, S. Liu, and G. Li. Real photographs denoising with noise domain adaptation and attentive generative adversarial network. In *CVPRW*, 2019.

[6] T. Ehret, A. Davy, J.-M. Morel, G. Facciolo, and P. Arias. Model-blind video denoising via frame-to-frame training. In *CVPR*, pages 11369–11378, 2019.

[7] S. Cha, T. Park, and T. Moon. GAN2GAN: Generative Noise learning for blind image denoising with single noisy images. *arXiv preprint arXiv:1905.10488,* 2019.

[8] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603,* 2018.

[9] J. Anaya and A. Barbu. RENOIR–A dataset for real low-light image noise reduction. *Journal of Visual Communication and Image Representation,* 51:144–154, 2018.

[10] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, pages 1692–1700, 2018.

[11] A. Krull, T.-O. Buchholz, and F. Jug. Noise2Void-learning denoising from single noisy images. In *CVPR*, pages 2129–2137, 2019.

[12] P. Hermosilla, T. Ritschel, and T. Ropinski. Total denoising: Unsupervised learning of 3D point cloud cleaning. *arXiv preprint arXiv:1904.07615,* 2019.

[13] J. Batson and L. Royer. Noise2Self: Blind denoising by self-supervision. *arXiv preprint arXiv:1901.11365,* 2019.

[14] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila. Noise2Noise: Learning image restoration without clean data. In *ICML,* pages 2965–2974, 2018.

[15] S. Niklaus and F. Liu. Context-aware synthesis for video frame interpolation. In *CVPR*, pages 1701–1710, 2018.

[16] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[17] M. K. Hossen and S. H. Tuli. A surveillance system based on motion detection and motion estimation using optical flow. In *International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 646–651, 2016.

[18] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015.

[19] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, pages 2462–2470, 2017.

[20] T.-W. Hui, X. Tang, and C. Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, pages 8981–8989, 2018.

[21] K. Zhang, W. Zuo, and L. Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing,* 27(9):4608–4622, 2018.

[22] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Joint pattern recognition symposium*, pages 214–223, 2007.

[23] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, pages 4161–4170, 2017.

[24] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018.

[25] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, pages 3354–3361, 2012

[26] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision,* 92(1):1–31, 2011.

[27] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, pages 611–625, 2012.

[28] D. Kesrarat and V. Patanavijit. Verification of video reconstruction using bilateral-reverse directional global based optical flow over non-Gaussian noise. *International Journal of Simulation-Systems, Science & Technology,* 19(3):4.1–4.6, 2018.

[29] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pages 231–236, 1993.

[30] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision,* 127(8):1106–1125, 2019.

[31] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *ECCV*, pages 286–301, 2016.

[32] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging,* 3(1):47–57, 2016.

[33] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980,* 2014.

[34] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI* pages 234–241, 2015.

[35] C. Liu and D. Sun. A Bayesian approach to adaptive video super resolution. In *CVPR*, pages 209–216, 2011.

[36] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016.

[37] M. Tassano, J. Delon, and T. Veit. FastDVDnet: Towards real-time video denoising without explicit motion estimation. *arXiv preprint arXiv:1907.01361,* 2019.

[38] S. Yu, B. Park, and J. Jeong. PoSNet: 4x video frame interpolation using position-specific flow. In *ICCVW*, 2019.

[39] E. Agustsson and R. Timofte: NTIRE 2017 challenge on single image super-resolution: dataset and study. In *CVPRW*, 2017.

[40] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black: A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[41] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poisson-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Tansactions on Image Processing*, 17(10):1737–1754, 2008.

[42] M. Maggioni, G. Boracchi, A. Foi, K. Egiazarian. Video denoising, deblocking and enhancement through separable 4-D nonlocal spatiotemporal transforms. *IEEE Tansactions on Image Processing*, 21(9):3952–3966, 2012.